The Rich Transcription Fall 2003 (RT-03F) Evaluation Plan

1 INTRODUCTION

The goal of this document is to define the evaluation tasks, performance measures, and test corpora to support the 2003 Rich Transcription Fall (RT-03F) evaluation. Rich Transcription (RT) is broadly defined to be a fusion of speech-to-text (STT)[1] technology and metadata extraction (MDE) technologies which will provide the basis for the generation of more usable transcriptions of human-human speech for both humans and machines. This (Fall) evaluation is a follow-on adjunct to the main (Spring) RT-03 evaluation, which was the second evaluation in a series intended to support the research and development of individual RT transcription and extraction components. This series provides the evaluation mechanisms to support DARPA's Effective, Affordable, Reusable Speech-to-text (EARS) Program.[2] Note, however, that in addition to EARS contractors, this evaluation is open to all interested volunteers. All participants will be required to attend the RT-03 Fall Workshop which will follow the evaluation.

The RT-03F evaluation will focus on areas that were deferred from the main Spring evaluation. Evaluation will be supported for six tasks:

Edit Word Detection

Filler Word Detection

IP Detection

SU Boundary Detection

Speaker Attributed STT

RT03 Rich Transcription

The RT-03F evaluation will be limited to English language only.

2 BACKGROUND

Beginning in the early 1980's, evaluation of automatic speech recognition (ASR) stabilized on the current performance measure of word error rate (WER). This measure scores ASR performance using a case-less lexicalized form of ASR output known as the "standard normalized orthographic representation" (SNOR) format.[3] The WER is defined as the sum of all ASR output token errors divided by the number of scoreable tokens in a reference transcription of the test data. There are three types of errors, these being tokens that are missed (deletion errors), inserted (insertion errors), and incorrectly recognized (substitution errors).[4]

While the traditional STT evaluations have helped to provide a mechanism for evaluating word accuracy, it is clear that words alone are insufficient in formulating a transcription of speech which is readable by humans and understandable by machines. A verbatim transcription of the speech stream into a string of lexical tokens yields a transcript that is often extremely difficult to understand. This is because spoken language is much more than just a string of lexical tokens. It contains information about the speaker, prosodic cues to the speaker's intent, and much more. Spoken language also contains disfluencies, which speakers correct and which textual renderings should delete. All of this makes the task of rendering spoken language into text a great challenge, especially with less-than-perfect ASR performance.

Solving these problems is the challenge that the EARS program takes as its objective and what the RT evaluation series seeks to assess – namely to develop technology that transforms spoken language into a form that is maximally informative. This requires new approaches to acoustical modeling and insightful models of disfluencies, dialogue and other relevant speaker behaviors.

3 THE RT-03F TASKS

The RT-03F evaluation includes six tasks. The first three tasks, edit word detection, filler word detection and interrupt point (IP) detection, are targeted at finding disfluent speech. The SU boundary detection task is designed to find the boundaries between SUs (defined below). The speaker attributed STT task associates speaker identity with the words spoken. The sixth task, RT03 rich transcription, is an integration of the other tasks.

3.1 MDE TASKS

Metadata extraction is a new research area in speech recognition, and one that is at an early stage of conceptual development. These metadata tasks, and the evaluations of them, are defined to be as independent of each other as possible.

3.1.1 DISFLUENCY TASKS

For RT-03F, three disfluency-related tasks are defined: edit word detection, filler word detection and IP detection. In EARS, disfuencies are portions of speech in which a speaker's utterance is not complete and fluent but that the speaker corrects, repeats, or abandons. Disfluencies are fully discussed and explained in the Simple MDE

[1] formerly known as automatic speech recognition (ASR)

[2] The EARS research effort is dedicated to developing powerful new speech transcription technology that provides substantially richer and more accurate transcripts than are currently possible. The research focus is on natural, unconstrained speech from broadcasts and telephone conversations in a number of languages. The program objective is to create core enabling technology suitable for a wide range of advanced applications.

[3] Since some languages' written forms are not word-based, this concept has been extended to cover lexemes – a representation of a written unit of meaning within a language. Thus, this document frequently refers to lexemes, lexical tokens, or tokens rather than words. For English, these terms may be treated more or less equivalently.

[4] Underlying the tabulation of errors is a requirement to align the tokens in the system output transcript with the tokens in the reference transcript. Traditionally, this has been done using a dynamic programming algorithm that searches for an alignment that minimizes the WER.

Annotation Specification[5]. The EARS motivation to detect disfluencies is to enable "clean up" of rendered text. Portions of disfluencies can be removed, or processed another way, is improve the readability of transcribed texts.

The two disfluency types, edits and fillers, have similar structures but they are independent speech events: filler disfluencies can occur anywhere within edit disfluencies. Thus, their detection has been divided into separate tasks.

There is a common structure to disfluencies. They consist of a DEPOD, an interruption point and a correction. The ordering of these elements and their presence defines the type of disfluency. Disfluencies are often nested, however the EARS program has decided to address only the top most level.

- A DEPOD is an EARS neologism defined as the DEletable Part Of a Disfluency. It is the speaker's initial attempt at an utterance. DEPODS are candidates for deletion (or other types of special handling) in the production of a rich transcript.
- An interruption point (IP) occurs at the point were there is a discontinuity between fluent and non-fluent speech and is a prosodic phenomenon.
- The correction consists of the portion of the utterance that has been repaired and is fluent. Correction portions are not always present.

Edit disfluencies have a DEPOD, one or more IPs and optionally a correction. There are four edit disfluency subtypes: repetitions, revisions, restarts and complex. Complex edits are nested disfluencies and have multiple IPs. IPs for edits are on the right edge, (the end), of the DEPOD, (and within the DEPOD for complex edits).

Filler disfluencies exhibit a DEPOD and an IP. Unlike the edit disfluencies, the IP is on the left edge, (the beginning), of the DEPOD. There are three subtypes of fillers defined by the Simple MDE Annotation spec.: filled pauses, discourse markers and explicit editing terms.

3.1.1.1 Edit Word Detection

The edit word detection task is to identify the DEPODs of edit disfluencies. The system's detection of edit disfluency subtypes is optionally.

3.1.1.2 Filler Word Detection

The filler word detection task is to identify the DEPODs of filler disfluencies. The system's detection of filler disfluency subtypes is optionally.

3.1.1.3 IP Detection

The IP detection task is to identify interruption points in speech.

There are three subtypes of IPs that are derived from the disfluency they are part of. They are:

- Edit – IPs from edit disfluencies
- Filler – IPs from filler disfluencies
- Filler&edit – IPs that occur between edit and filler disfluencies.

The system's detection of IP subtypes is optional.

3.1.2 SU Boundary Detection

The SU[6] boundary task is to identify the end-points of sentence-like groups of words in STT transcripts. Such points could influence punctuation and capitalization (or other types of special handling) in the production of a rich transcript. For RT-03F, four types of SUs are defined: statement, question, backchannel and incomplete. SU type detection is an optional output for SU Boundary Detection.

3.2 Integrated MDE and STT Tasks

The Rich Transcription Evaluation series' primary goal is to develop enriched transcriptions composed of both STT and MDE annotations. RT-03F defines two integrated MDE/STT tasks: Speaker Attributed STT and RT03 Rich Transcription.

3.2.1 Speaker Attributed STT

Speaker Attributed STT conceptually combines STT with speaker labeling. The speaker-labeling task is to annotate each STT output word with the correct speaker label, so that all words spoken by the same speaker have the same speaker label. (The name of the speaker is not required and will not be evaluated. Rather, it is the consistency of speaker labeling that is important – all of the words spoken by a speaker should have the same, arbitrary speaker label.)

3.2.2 RT03 Rich Transcription

The RT03 Rich Transcription task[7] combines STT with three metadata tasks: edit word detection, filler word detection, SU boundary detection and speaker attributed STT.

Systems shall produce STT plus metadata information, edit and filler DEPOD regions, SU boundaries, and speaker label for each word.

4 Performance Measures

Separate performance measures are defined for each of the major EARS tasks. Evaluation will be performed separately for each file and for each channel within a file then aggregated over the test set. RT-03F supports three performance measurement tools, two accepts input in RTTM format (see Appendix A) and one accepts input in RT-XML format[8] (see Appendix B). (A conversion tool will be provided to convert between the two different formats.) Participants may choose either format for their system to output.

[5] http://macears.ll.mit.edu/macears_docs/data/SimpleMDE_V5.0.pdf

[6] SUs have been variously defined as "slash units", "sentence units", "semantic units" and "structural units"

[7] The task was named RT03 Rich Transcription because the integrated task of Rich Transcription is likely to change over time as new metadata types are added to the rich transcript.

[8] RT-XML format represents metadata in terms of annotations on STT output tokens and thus requires STT output in addition to metadata output.

4.1 Evaluation Tools and File Formats

There are three scoring tools for the evaluation: *su-eval*, *df-eval* and *rteval*. *Su-eval* and *df-eval* require RTTM formatted files while *rteval* requires RT-XML. SU boundary detection is scored with either *su-eval* or *rteval*. The filler word detection, edit word detection and IP detection tasks are scored with either *df-eval* or *rteval*. The speaker attributed STT and RT-03 rich transcription tasks are scored with *rteval*.

4.2 Metadata Event Measures

4.2.1 Measures for Disfluency Tasks

4.2.1.1 Measure for Edit Word Detection

An overall edit word detection error score will be computed as the average number of misclassified reference tokens per reference edit token.

$$Error_{depod} = \frac{\left(\begin{array}{l}\#\text{ of depod ref tokens not covered by sys depods} \\ +\#\text{ of non - depod ref tokens covered by sys depods}\end{array}\right)}{\#\text{ of ref tokens in the ref depods}}$$

The formula is specifically tied to the reference word tokens. By doing so, STT deletion errors within DEPODs do not contribute errors, while STT insertions that are spuriously labeled as edits do contribute errors.

4.2.1.2 Measure for Filler Word Detection

An overall filler word detection error score will be computed as the average number of misclassified reference tokens per reference filler token.

$$Error_{depod} = \frac{\left(\begin{array}{l}\#\text{ of depod ref tokens not covered by sys depods} \\ +\#\text{ of non - depod ref tokens covered by sys depods}\end{array}\right)}{\#\text{ of ref tokens in the ref depods}}$$

Like edit word detection error, the formula is specifically tied to the reference word tokens. By doing so, STT deletion errors within DEPODs do not contribute errors, while STT insertions that are spuriously labeled as edits do contribute errors.

4.2.1.3 Measure for IP Detection

The overall IP error rate will be simply the average number of missed IP detections and falsely detected IPs per reference IP:

$$Error_{IP} = \frac{\left(\#\text{ of missed IP's} + \#\text{ of false alarm IP's}\right)}{\#\text{ of ref IP's}}$$

4.2.2 Measure for SU Boundary Detection Task

An overall SU error score will be computed as the average number of missed SU boundary detections and falsely detected SU boundaries per reference SU:

$$Error_{SU} = \frac{\left(\#\text{ of missed SU's} + \#\text{ of false alarm SU's}\right)}{\#\text{ of ref SU's}}$$

4.3 Integrated MDE and STT Performance Measures

4.3.1 Measure for Speaker Attributed STT Task

An overall error score for the speaker attributed STT task will be computed as the number of errorful **lexemes**, expressed as a fraction of reference **lexemes**: An errorful lexeme is a lexeme that is undetected (deleted), or spuriously detected (inserted), or for which the speaker is incorrectly recognized.

4.3.2 Measure for RT03 Rich Transcription Task

An overall error score for the RT03 rich transcription task will be computed as the number of errorful rich transcription output tokens, expressed as a fraction of the reference rich transcription output tokens. An errorful token is a token that is undetected (deleted), or spuriously detected (inserted), or incorrectly recognized.

An errorful rich transcription output token declared if…

- the reference token lexical identity does not match the system token lexical identity
- the reference token disfluency type is marked "edit" or "filler" and the the system token disfluency type is marked "none" (or vice versa). Note that "edit" vs. "filler" confusions do not cause substitution errors for the TER.
- the reference token su_boundary value does not match the system token su_boundary value
- the reference speaker label does not match the mapped system speaker label

4.4 Scoreable Tokens

The edit word detection, filler word detection, speaker attributed STT, and RT03 rich transcription error measures count scorable tokens. A scoreable token is defined to be all subtypes of the RTTM LEXEME class. The subtypes are: 'lex', 'fp', 'frag', 'un-lex', 'for-lex', 'alpha', 'acronym', 'interjection', 'propername' and 'other'. All scoreable tokens count towards the metrics' denominator.

5 Corpora Resources

5.1 Evaluation Training Data

Any released broadcast news or CTS LDC corpora may be used for the training and development of the MDE tasks. Note that **all** material used in **any** way for training and development for the broadcast news recognition tasks must predate the test epoch (February 2001) as specified in Section 7.1.2.

The LDC has begun annotating data to the Simplified MDE Annotation Specification. At the time of this document, July 30, 2003, the LDC has agreed to release 40 Switchboard I conversations that were annotated to the Meteer et al. specification and the Fall 2002 Dry Run Data Broadcast news data set.

Consult the MACEARS web site, **http://ears.ll.mit.edu/**, for currently released data.

5.2 DRY RUN EVALUATION/DEVELOPMENT TEST DATA

The dry run test corpora will consist of the RT03-set2 data. The Broadcast News file names are:

20010206_1830_1900_ABC_WNT
20010221_1830_1900_NBC_NNW
20010225_0900_0930_CNN_HDL

The Conversational Telephone Speech files are:

fsh_60386	fsh_60398	fsh_60441	fsh_60477	fsh_60568
fsh_60613	fsh_60668	fsh_60682	fsh_60784	fsh_60817
fsh_60818	fsh_60844	fsh_60874	fsh_61113	fsh_61130
fsh_61148	fsh_61225	fsh_61228	sw_45104	sw_45237
sw_45481	sw_45626	sw_45837	sw_45856	sw_45973
sw_46028	sw_46168	sw_46455	sw_46565	sw_46671
sw_46732	sw_46938	sw_47038	sw_47073	sw_47175
sw_47282				

5.3 RT-03 FALL EVALUATION TESTING DATA

The RT-03 Fall evaluation test corpora will come from the RT-03-set1 data set. The Broadcast News filenames are:

20010217_1000_1030_VOA_ENG
20010220_2000_2100_PRI_TWD
20010228_2100_2200_MNB_NBW

The Conversational Telephone Speech files are:

fsh_60262	fsh_60354	fsh_60416	fsh_60463	fsh_60493
fsh_60549	fsh_60571	fsh_60593	fsh_60627	fsh_60648
fsh_60650	fsh_60720	fsh_60732	fsh_60797	fsh_60862
fsh_60885	fsh_61039	fsh_61192	sw_45097	sw_45142
sw_45355	sw_45454	sw_45586	sw_45654	sw_45713
sw_45727	sw_45819	sw_46140	sw_46412	sw_46512
sw_46615	sw_46677	sw_46789	sw_46868	sw_47346
sw_47411				

The RT-03 Fall evaluation is not a "blind" evaluation. Participants have had access to transcribed data since the RT-03 Spring evaluation. Further, the now-designated evaluation set was intended to be the development test set. As such, participants began working in earnest developing systems until the community realized the wrong data set was annotated for the development test set. Rather than delay the Fall evaluation, the development and evaluation data sets were swapped and participants were instructed to discontinue work with the evaluation data set. The following rules governing the use of the RT-03 data were instituted on July 7, 2003 per the EARS executive board.

Individual researchers who are likely to participate in the RT-03F evaluation should only use the dev set (RT-03 set 2, see Section 5.2) and should cease using evaluation data set (RT-03 set 1). Participants must disclose in their system description how they used the Fall evaluation data set (RT-03 set 1), if at all, prior to the July 7th decision to swap test sets. The disclosure is intended to publicly acknowledge usage of the test set. Researchers who are not intending to participate in RT-03F can use sets 1 and 2 as they see fit.

6 EVALUATION CONDITIONS

There are many different conditions under which system performance may be evaluated. This section identifies those conditions for which performance will be computed and, of those, which are to be designated as the "primary" evaluation conditions.

The following list of evaluation conditions apply to all of the five RT-03 Fall Evaluation tasks.

- **Language**:
  - English only
- **Domain**:
  - Broadcast news and conversational telephone speech. Participants may build systems to address either or both domains.
- **Input**:
  - Speech input. Any desired fully-automatic signal processing approaches may be employed (including the use of a site developed STT system).
  - Speech plus the reference transcriptions: The function of this evaluation condition is to serve as a perfect STT control condition. Thus, the system inputs will be RTTM formatted files derived from the reference RTTM file and placed in the 'inputs' directory (described in section 7.2 below) of the evaluation corpus. The derived RTTM files will contain only 'LEXEME' RTTM records with the speaker's identity expunged, (replaced by <NA>), and the LEXEME subtypes 'alpha', 'acronym', 'interjection', 'propername', and 'other' mapped into the 'lex' subtype.

7 PARTICIPATION INSTRUCTIONS

Participation is encouraged for all those who are interested in one or more of the RT-03 Fall tasks. All participants must, however, agree to completely process all of the data for at least one task. This means that, at a minimum, the speech-input-only processing conditions must be implemented. Participants have the freedom to implement systems for either or both domains, Broadcast News, or Conversation Telephone Speech.

As a condition of participation, all sites must agree to make their submissions (system output, system description, and ancillary files) available for experimental use by other research sites. Further, submission of system output to NIST constitutes permission on the part of the site for NIST to publish scores and analyses for that data including explicit identification of the submitting site and system.

7.1 PROCESSING RULES

7.1.1 RULES THAT APPLY TO ALL EVALUATIONS

All processing for all tasks must be fully automatic. No manual intervention of any kind is permitted. Adaptation is permitted for all tasks, however it too must be fully automatic. The only exemption from the automatic processing restriction is for the reference text condition. Participants who use the reference text condition can manually add pronunciations to their dictionaries to enable forced alignment of the out-of-vocabulary terms. Participants cannot use the lexical knowledge gained to modify their "speech-input" only systems.

Systems will be provided with recorded waveform files and an index file specifying the speech files and regions within them to be processed. Conversational telephone speech test data will be provided in 2-channel files, and both channels must be processed. Broadcast news speech test data will be presented in single channel files. Each conversation and each news broadcast excerpt to be processed will be presented in a separate file.

While entire broadcast and conversation files will be distributed, only the material specified in the UEM test index file for the experiment to be run is to be processed. Material outside of the times specified in the UEM test index file is not to be used in any way (e.g., for adaptation).

7.1.2 Additional rules for processing Broadcast News

News-oriented material (audio, textual, etc.) generated during or after the test epoch beginning February 01, 2001 **may not be used in any way for system development or training.** Broadcast news material must be processed in the chronological order of the date/time of the original broadcast. Although automatic adaptation may be performed using previously-processed material, systems may not "look ahead" in time at later recordings. Hence, processing must be complete on a particular broadcast news test file before moving on to the next file. Any form of within-file adaptation is however permitted and systems may look backwards in time at previously-processed files. The show identity and original broadcast date are allowable side information that systems may use. Therefore, systems may make use of show-dependent models.

7.1.3 Additional rules for processing conversational telephone speech

Conversational telephone speech may be processed in any order and any form of automatic within-conversation and cross-conversation adaptation may be employed. No side information is provided for telephone conversations. No manual or automatic segmentation will be provided, although systems may make use of segmentation outputs donated from other sites.

7.2 Data Formats

7.2.1 Test data

For practicality, the recorded waveform files to be processed will be distributed on CD-ROM and the corresponding indices, annotations, and transcripts will be made available via the Web or FTP using an identical directory structure. After the evaluation, system outputs will be released in this structure as well.

Directory	Description
indices/	index files containing the list of files and times to be processed for particular experiments
audio/	audio files
input/<EXP-ID>/	ancillary data including reference annotations for various experiments – must be used in accordance with instructions for that experiment
output/<EXP-ID>/	system output submissions – will be made available as received for integration tests [9]
reference/	reference transcripts, annotations, and MS-wav files for post-evaluation scoring and analyses

Note: EXP-ID specifies a unique identifier for each experiment and is defined in 7.3.1.

For clarity, the "audio/" and "reference/" directories are subdivided into <DATA>/<LANG>/<TYPE> subdirectories:

Where:

<DATA> is either [dev03f|eval03f]

<LANG> [english]

<TYPE> is either [bnews|cts]

The "indices/" directory contains a set of UEM test index files specifying the waveform data to be evaluated for each EXP-ID condition supported in this evaluation as described in 7.3.1 and are named <EXP-ID>.uem with the special site code "expt". A separate .uem file will be provided for each experiment for each supported <DATA>, <LANG>, and <TYPE>. Only the waveform data specified in these files should be processed for the given experimental condition. Corresponding ancillary data for some control conditions is given in the "input/" directory under subdirectories with the same EXP-ID. These files contain new-line-separated records and whitespace-separated fields of the form:

<FILE><SP><CHANNEL><SP><BEGIN-TIME><SP><END-TIME><NEW-LINE>

where,

<SP> is whitespace

<FILE> specifies the path and filename of the waveform file to be processed

<CHANNEL> specifies the channel within the waveform file to be processed

<BEGIN-TIME> and <END-TIME> specify the time region within the specified file to be processed.

For example:

The index file expt_03_stt10x_dev_eng_cts_spch_1.uem will contain:

.

.

audio/dev/english/cts/sw_47620.sph 0 0 291.34

.

.

7.2.2 MDE Output Format

7.2.2.1 RTTM Format

See Appendix A for a description of the RTTM format. Each RTTM file corresponds to a single source file in the test.

7.2.2.2 RT-XML Format

See Appendix A for a description of the RT-XML format. Each RTTM file corresponds to a single source file in the test.

[9] However, no data regarding the Progress tests will be posted.

7.2.3 System Description

For each test ran (for each unique EXP-ID), a brief description of the system (algorithms, data, configuration) used to produce the system output must be provided along with your system output. If multiple system runs are submitted for a particular experiment with different systems/configurations, explicitly designate one run as the primary system and the others as contrastive systems in the system description. This information is to be recorded in a file named:

<EXP-ID>.txt

(where EXP-ID is defined in Section 7.3.1)

and placed in the "output" directory alongside the similarly-named directories containing your system output. This file is to be formatted as follows:

1. EXP-ID = <EXP-ID>
2. Primary: yes | no
3. System Description:

   [brief technical description of your system; if a contrastive test, contrast with primary system description]

4. Training:

   [list of resources used for training; for STT, be sure to address acoustic and LM training, and lexicon]

5. References:

   [any pertinent references]

7.3 Submission Instructions

7.3.1 Submission Experiment Codes

The output of each submitted experiment must be identified by the following code as specified above.

EXP-ID =
<SITE>_<YEAR>_<TASK>_<DATA>_<LANG>_
<TYPE>_<COND >_<SYSID>_<RUN>

Where,

SITE ::= expt | bbn | bbnplus | cu | elisa | clips | sri | sriplus | ibm | mitll | ms | pan | ...

(The special SITE code "expt" is used in the EXP-ID-based filename of the UEM test index files under the "indices/" directory to list the test material for a particular experiment and in the EXP-ID-based subdirectory name under the "input/" directory to indicate ancillary data to be used in certain control condition experiments.)

YEAR ::= 03f

For the RT-03 Fall Evaluation, these are:

TASK ::= ewd | fwd | ipd | subd | disf | 03rt | data

where,

ewd = edit word detection

fwd = filler word detection

ipd = IP Detection

subd = SU Boundary Detection

sastt = Speaker Attributed STT

03rt = RT-03 Rich Transcription

data = a special TASK code be used to provide a directory for ancillary data such as common CTM files used over many MDE experiments. Please make sure to use increasing run numbers for this special experiment ID when making multiple submissions so that your ancillary data from earlier submissions is not over-written here at NIST

DATA ::= dev03f | eval03f

LANG ::= eng | arab | mand

RT-03F only includes English (eng) material and not Arabic (arab) or Mandarib (mand).

TYPE ::= bnews | cts

CONDITION ::= spch | ref

Where,

spch = audio input only

ref = audio input + reference transcript

(The "spch" [speech] condition is the primary condition of interest. The "ref" [reference] condition is provided as a control for perfect speech recognition and includes both the speech and reference transcript as input. The MDE tasks for this condition may make use of only the LEXEME entries in the supplied RTTM as defined in Section 6 "Evaluation Conditions".

SYSID ::= site-named string designating the system used

[This is intended so that we can differentiate between contrastive runs for the same condition. Therefore, a different SYSID should be created for runs where any manual changes were made to a particular system]

RUN ::= 1..n (with values greater than 1 indicating multiple runs of the same experiment/system)

[An incremental run number MUST be used for multiple submissions of any particular experiment with an identical configuration (due to a bug or runtime problem.) This should NOT be used to indicate contrastive runs. Instead, a different SYSID should be used. However, please note that ONLY the first run will be considered "official" and will be scored by NIST unless special arrangements are made with NIST. Please also note that submissions which reuse identical experiment IDs/run numbers from previous submissions will be automatically rejected.]

examples:

bbn_03f_ip_eval03f_eng_cts_spch_superreco1_1

sri_03f_sastt_eval03f_eng_bnews_ref_speakerid2_1

7.3.2 Submission directory structure

All system output submissions must be formatted according to the following directory structure:

output/<SYSTEM-DESCRIPTION-FILES>

output/<EXP-ID>/ <OUTPUT-FILES>

where,

<SYSTEM-DESCRIPTION-FILES> one per <EXP-ID> as specified in 7.2.3

<EXP-ID> is as defined in Section 7.3.1

<OUTPUT-FILES> are as defined in Section 7.2

Note: one output file must be generated for EACH input file as specified in the test index for the experiment being run. The output files are to be named so as to be identical to the input file basenames with the appropriate .ctm or .rttm filetype extension. For example, an STT output file for the speech waveform file sw_47620.sph must be named sw_47620.ctm and an MDE output file must be named sw_47620.rttm. When generated, these output files are to be placed under the appropriately-named EXP-ID directory on your system identifying the experiment run.

7.3.3 SUBMISSION PACKAGING AND UPLOADING

To prepare your submission, first create the previously-described file/directory structure. This structure may contain the output of multiple experiments, although you are free to submit one experiment at a time if you like. The following instructions assume that you are using the UNIX operating system. If you do not have access to UNIX utilities or ftp, please contact NIST to make alternate arrangements.

First change directory to the parent directory of your "output/" directory. Next, type the following command:

tar -cvf - ./output | gzip > <SITE>_<SUB-NUM>.tgz
where,

<SITE> is the ID for your site as given in Section 7.3.1

<SUB-NUM> is an integer 1 – n where 1 identifies your first submission, 2 your second, and so forth.

This command creates a single tar file containing all of your results. Next, ftp to jaguar.ncsl.nist.gov giving the username 'anonymous' and your e-mail address as the password. After you are logged in, issue the following set of commands, (the prompt will be 'ftp>'):

ftp> cd incoming
ftp> binary
ftp> put <SITE>_<SUB-NUM>.tgz
ftp> quit

You've now submitted your recognition results to NIST. The last thing you need to do is send an e-mail message to Audrey Le at audrey.le@nist.gov to notify NIST of your submission. The following information should be included in your email:

1) The name of your submission file

2) A listing of each of your submitted experiment IDs

3) e.g.,
```
Submission: bbnplus_1 <NL>
Experiments: <NL>
bbnplus_03_stt10x_eval03_eng_cts_spch
_superreco1_1<NL>
bbnplus_03_stt10x_eval03_eng_cts_spch
_superreco2_1 <NL>
.
.
```

Note that submissions received after the stated due dates FOR ANY REASON will be marked late. So, please submit your files in time for us to deal with any transmission/formatting problems that might occur well before the due date if possible.

8 SCHEDULE

The evaluation schedule below is accurate as of July 30, 2003. Please consult the live version of the schedule at http://macears.ll.mit.edu/macears_docs/ears-schedule.txt for any late-breaking changes.

1 Oct - NIST releases eval data to sites

15 Oct - System output due at NIST

22 Oct - NIST releases scored results

5 Nov - Slides for notebooks due

13-14 Nov - RT03F Workshop/PI Meeting

Please note that the stated dates are hard deadlines. All late submissions will be marked as such and given the tight schedule, severely late submissions may not be scored at all prior to the workshops.

9 WORKSHOPS

The evaluation will be followed by the Rich Transcription 2003 Fall (RT-03F) Workshop. The workshop is open to all participants. Information regarding workshop logistics and registration will be posted at a later date in email.

Appendix A: RTTM File Format Specification

This description has been excerpted from RTTM-format-v11.doc[10]. There are four general object categories to be represented. They are STT objects, MDE objects, source (speaker) objects, and "segments".[11] Each of these general categories may be represented by one or more types and subtypes, as shown in table 1.

Table 1 Rich Text object types and subtypes

Type	Subtypes
SEGMENT	**eval**, or (none)
STT types:	
LEXEME	**lex**, **fp**, **frag**, **un-lex**[12], **for-lex**, **alpha**[13], **acronym**[13], **interjection**[13], **propername**[13], and **other**
NON-LEX	**laugh**, **breath**, **lipsmack**, **cough**, **sneeze**, and **other**
NON-SPEECH	**noise**, **music**, and **other**
MDE types:	
FILLER	**filled_pause**, **discourse_marker**, **explicit_editing_term**, and **other**
EDIT	**repetition**, **restart**, **revision**, **simple**, **complex**, and **other**
IP	**edit**, **filler**, **edit&filler**, and **other**
SU	**statement**, **backchannel**, **question**, **incomplete**, **unannotated**, and **other**
CB	**coordinating**, **clausal**, and **other**
A/P	(none)
SPEAKER	(none)
Source information:	
SPKR-INFO	**adult_male**, **adult_female**, **child**, and **unknown**

Each of these objects (except for **SEGMENT**) represents an EARS research target. And, except for the static speaker information object [**SPKR-INFO**], each object exhibits a temporal extent with a beginning time and duration. (The duration of interruption points [**IP**] and clausal boundaries [**CB**] is zero by definition.)

These objects are represented individually, one object per record, using a flat record format with object attributes stored in white-space separated fields. The format is shown in table 2.

Table 2 Object record format for EARS objects

Field 1	2	3	4	5	6	7	8	9
type	File	chnl	tbeg	tdur	ortho	stype	name	conf

where

`file` is the waveform file base name (i.e., without path names or extensions).

[10] The latest RTTM format document can be found at the URL 'http://www.nist.gov/speech/tests/rt/rt2003/fall/index.htm'.

[11] Although the "segment" is an artificial construct, it is important because it is produced by LDC to provide a modicum of temporal organization in the annotation.

[12] Un-lex is also used to tag words that are infected with or affected by laughter.

[13] This subtype is an optional addition to the previous set of lexeme subtypes which is provided to supplement the interpretation of some lexemes.

`chnl` is the waveform channel (e.g., "**1**" or "**2**").

`tbeg` is the beginning time of the object, in seconds, measured from the start time of the file.[14] If there is no beginning time, use `tbeg` = "**<NA>**".

`tdur` is the duration of the object, in seconds.[4] If there is no duration, use `tdur` = "**<NA>**".

`stype` is the subtype of the object. If there is no subtype, use `stype` = "**<NA>**".

`ortho` is the orthographic rendering (spelling) of the object for STT object types. If there is no orthographic representation, use `ortho` = "**<NA>**".

`name` is the name of the speaker. `name` must uniquely specify the speaker within the scope of the `file`. If `name` is not applicable or if no claim is being made as to the identity of the speaker, use `name` = "**<NA>**".

`conf` is the confidence (probability) that the object information is correct. If `conf` is not available, use `conf` = "**<NA>**".

This format, when specialized for the various object types, results in the different field patterns shown in table 3.

Table 3 Format specialization for specific object types

Field 1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
type	*file*	*chnl*	*tbeg*	*tdur*	*ortho*	*stype*	*name*	*conf*
SEGMENT	file	chnl	tbeg	tdur	**<NA>**	**eval** or **<NA>**	name or **<NA>**	conf or **<NA>**
LEXEME **NON-LEX**	file	chnl	tbeg	tdur	ortho or **<NA>**	stype	name	conf or **<NA>**
NON-SPEECH	file	chnl	tbeg	tdur	**<NA>**	stype	**<NA>**	conf or **<NA>**
FILLER **EDIT** **SU**	file	chnl	tbeg	tdur	**<NA>**	stype	name	conf or **<NA>**
IP **CB**	file	chnl	tbeg	**<NA>**	**<NA>**	stype	name	conf or **<NA>**
A/P **SPEAKER**	file	chnl	tbeg	tdur	**<NA>**	**<NA>**	name	conf or **<NA>**
SPKR-INFO	file	chnl	**<NA>**	**<NA>**	**<NA>**	stype	name	conf or **<NA>**

[14] If `tbeg` and `tdur` are "fake" times that serve only to synchronize events in time and that do not represent actual times, then these times should be tagged with a trailing asterisk (e.g., `tbeg` = **12.34*** rather than **12.34**).

Appendix B: RT-XML File Format Specification

This appendix will be filled in after the RT-XML Format Specification is completed.